Japanese Patent Laid-open No. HEI10-333902 A

Publication date : December 18, 1998

Applicant : N II C JOHO SYST:KK

Title : COMPUTER SYSTEM WITH TAMPER CHECKING FUNCTION

(57) [Abstract]

[Problems to be Solved] At boot-up of a computer, it is checked whether or not an OS, an OS loader, and other files to be checked are tampered, and after confirmation of their safety, the OS, the OS loader and the application are started.

[Solving Means] A boot program 2 stored in a ROM 1 reads first stored information 41 stored in an auxiliary storage 4 not via an OS after initialization of a BIOS, and reads, in accordance with read contents, a file to be checked 42 without involving the OS. These read contents and second stored information stored in the ROM 1 are used to check presence or absence of the file to be checked. After confirming that there has occurred no tamper, an OS loader 43 is called to start up an OS 44.

[Scope of Claims for Patent]

[Claim 1] A computer system having a tamper checking function comprising in which a boot program in a ROM starts up at a boot-up, the boot program starts an OS loader in an auxiliary storage, and the OS loader starts an OS in the auxiliary storage, wherein

the boot program comprises:

a stored information reading unit that reads a first stored information stored in the auxiliary storage;

a file to be checked reading unit that reads an arbitrary number of files to be checked from the auxiliary storage based on a description read in the first stored

information; and

a tamper detecting unit that checks presence or absence of tamper in each of the files to be checked based on second stored information, the first stored information which is read and each of the files to be checked in the boot program.

[Claim 2] A computer system, wherein

a digital signature for each of files to be checked is stored in the first stored information,

a key for verifying the signature is stored in the second stored information, and

the tamper detecting unit according to claim 1 checks presence or absence of tamper of each of the files to be checked from the digital signature and the key for verifying the signature.

[Claim 3] A computer system, wherein an OS loader stored in the auxiliary storage is, as one of the files to be checked, subjected to tamper check at boot-up.

[Claim 4] A computer system, wherein an OS file stored in the auxiliary storage is, as one of the files to be checked, subjected to tamper check at boot-up.

[Claim 5] A computer system, wherein

the first stored information includes information for tamper check of the first stored information,

the tamper detecting unit according to claim 1 has a function of checking tamper of the first stored information.

[Claim 6] A computer system, wherein

a compressed value which is obtained by compressing each of the files to be checked by a one-way function and a digital signature of the first stored information are stored in the first stored information,

a key for verifying the signature is stored in the second stored information, and

the tamper detecting unit checks presence or absence of tamper of the first stored information by using the digital signature and the key for verifying the signature, and checks presence or absence of tamper of each of the files to be checked by using the compressed value.

[Claim 7] A computer system, wherein

the boot program comprises a stored information obtaining unit that obtains the second stored information from an input device, and

the second stored information is obtained by inputting of a user.

[Claim 8] The computer system according to claim 7, wherein

a compressed value obtained by compressing each of the files to be checked and the second stored information by a one-way function, and a compressed value obtained by compressing the first stored information and the second stored information by a one-way function are stored in the first stored information,

the tamper detecting unit calculates the compressed value again by using the second stored information obtained by the stored information obtaining unit according to claim 7 and compares a resultant value with the compressed value stored in the first stored information to check presence or absence of tamper of each of the files to be checked and the first stored information.

[Claim 9] A computer system, wherein

the second stored information is included in an external storage,

the boot program has a second stored information reading unit that obtains the second stored information from the external storage, and

the external storage comprises an anti-tamper unit of the second stored information.

[Claim 10] A computer system in which the second stored information is stored in the

external computer, wherein

the external computer comprises:

an anti-tamper unit of the second stored information; and

a tamper detection calculation unit that carries out calculations necessary for tamper detection using the second stored information,

the boot program comprises communicating unit with the external computer,

the tamper detecting unit according to claim 1 checks presence or absence of tamper of each of files to be checked by using the first stored information, each of the files to be checked, and using the tamper detection calculating unit at the external computer via the communicating unit with the external computer.

[Claim 11] A computer system, wherein

a tamper detecting program with a tamper detecting function according to claim 1 is stored in a storage having a write protect function, and

the boot program comprises an executing unit of the tamper detecting program.

[Claim 12] A computer system, wherein

a tamper detecting program with a tamper detecting function according to claim 1 is stored in the auxiliary storage,

a value obtained by compressing the tamper detecting program by a one-way function is stored in the ROM in advance,

the boot program comprises:

a tamper detecting program checking unit that checks presence or absence of tamper of the tamper detecting program per se by compressing the tamper detecting program by a one-way function and comparing a compressed value with the compressed value stored; and

a tamper detecting program executing unit that executes the tamper detecting program.

[Claim 13] A computer system, wherein

a tamper detecting program with a tamper detecting function according to claim 1 is stored in the auxiliary storage,

the OS loader stored in a boot sector of the auxiliary storage is stored in advance in a portion other than the boot sector of the auxiliary storage, and

an alternate OS loader is stored in a boot sector of the auxiliary storage, the alternate OS loader comprising:

a tamper detecting program executing unit that executes reading of the tamper detecting program from the auxiliary storage; and

an OS loader executing unit that executes reading of the OS loader stored.

[0010]

[Problems to be Solved by the Invention] The problems of the conventional art are problems of operations.

[0011] This is because, as mentioned in the first example of the conventional art, for safer operations, a computer system is booted up by physically unrewritable safe mediums before check is executed. However, actually almost computer users usually store OS or application programs in hard disks. A hard disk is not physically unrewritable medium and is likely to be tampered. Therefore, a computer system is booted up by a floppy disk which has OS and tamper checking program stored therein and which is write protected, and after checking OS and OS loader in the hard disk, the hard disk has to be booted up again. This bothers uses in operation and requires unnecessary latency time of system boot-up.

[0012] An object of the present invention is to check with reliability presence or absence of tampered program to be executed at boot-up of the OS loader, Os or the computer under stress-free operation of users.

[0013] It is another object of the present invention to simplify the configuration of a device for realizing the first object.

[0014] It is still another object of the present invention to increase the speed of tamper detecting processing.

[0015] It is still another object of the present invention to improve maintainability of the whole system.

[0033]

[Embodiments of the Invention]

[1] Descriptions of Configuration

Next, embodiments of the present invention are described in detail with reference to the drawings.

[0034] Referring to Fig. 1, a first embodiment of the present invention comprises a controller 100 and an auxiliary storage 4.

[0035] The controller 100 comprises a ROM 1, a RAM 102 and a CPU 101 that executes programs stored in the ROM and the RAM.

[0036] A boot program 2 stored in the ROM 1 is a program of reading into the RAM 102 an OS loader 43 stored in the auxiliary storage 4 after initializing peripheral devices including the auxiliary storage 4 to start up a basic I/O system (hereinafter referred to as BIOS). The present invention provides this boot program 2 with a tamper detecting function.

[0037] The auxiliary storage 4 includes an OS file 44 and an OS loader 43 for

reading and executing the OS file 44 into the RAM 102. In the present invention, the auxiliary storage 4 has an arbitrary number of files which are to be set as files to be checked 42 of which presence or absence of a tamper is checked at boot-up of the system, and setting data, information for tamper check, for example digital signature information are stored in first stored information 41.

[0038] The above-mentioned tamper detecting function 3 comprises a stored information reading unit 31, a file to be checked reading unit 32 and a tamper detecting unit 33.

[0039] The stored information reading unit 31 reads the first stored information 41 from the auxiliary storage 4. The time of performing this processing, in other words, the start-up of the boot program 2 is to be performed before the boot-up of the OS, and therefore, it is impossible to access files via the OS. Accordingly, access to the auxiliary storage 4 has to be performed only with a function of low level BIOS. For example, when the OS is MS-DOS of Microsoft Corporation, directory data and a file allocation table (hereinafter referred to as FAT) of the auxiliary storage 4 are referred to so as to obtain information on physical location of each file stored in the auxiliary storage 4. In other words, a file name and a corresponding entry point of the FAT are written in the directory data, and a physical location of a stored file can be obtained by going around the FAT from the entry point. Hence, once a file name of the first stored information 41 is fixed in advance, the file name is used to obtain the physical location of the stored file in the auxiliary storage 4, and this may be performed by using the BIOS. The same hold true with other OS, in which a procedure of obtaining a physical location of a stored file executed by the OS is obtained to read the position from the BIOS.

[0040] The file to be checked reading unit 32 obtains contents of the file to be

checked 42 from the read first stored information 41 to read the file to be checked 42 from the auxiliary storage 4. Also in the file to be checked reading unit 32, access to the storage has to be taken only by using a function of low level of BIOS. Like in the above-mentioned stored information reading unit 31, a physical location in the auxiliary storage 4 is obtained in the same method as that a used OS executes to read the position from the BIOS.

[0041] The tamper detecting unit 33 checks presence or absence of tamper of a file to be checked read from the auxiliary storage 4 by using check information read from the first stored information 41 and second stored information 34 of a boot program per se. For example, a digital signature for each file to be checked is stored as first stored information 41 and a key for signature check is stored as second stored information 34. The tamper detecting unit 33 uses the digital signature and the key for signature check to check presence or absence of tamper of the file to be checked.

[0042] The tamper detecting function 3 can be also adopted in detecting presence or absence of tamper of the OS loader 43. The OS loader is stored in a portion called boot sector in the auxiliary storage 4. The boot sector is located at a physically head portion of the auxiliary storage, or if it is divided in plural partitions, at a head portion of each of the partitions. A drive name of a boot sector to be checked is written in a box storing a file name of the first stored information 41, thereby to specify a physical position in the auxiliary storage 4. Further, if data for checking tamper of the boot sector, for example, a digital signature is stored, the tamper detecting unit 33 can be used to check presence or absence of tamper of the boot sector or of the OS loader 43 stored therein. More specifically, since the OS loader 43 is a part of the boot sector, the presence or absence of tamper of the boot sector and the presence or absence of tamper of the OS loader 43 are different. However, since information on record

format and the like is stored in boot sectors other than that of the program of the OS loader 43, the boot sectors also may be subjected to tamper check. If the tamper check is limited to only the program portion, for example, a field for storing the number of first skipping bites and the number of end ignored bites has only to be added to the first stored information 41. Thus, only a program part to be checked is specified.

[0043] The tamper detecting function 3 can be also adopted in detecting presence or absence of tamper of the OS file 44. The OS file 44 is here a generic name for a basic OS program, a setting file which the OS program reads and various drivers to be executed at boot-up of the OS. Take, for example, MS-DOS of Microsoft Corporation, where the basic programs has file names of "io.sys", "msdos.sys" and "command.com", setting files which the OS programs reads include files of "config.sys" and "autoexec.bat" and the drivers, which are described in the setting files, are executed at the boot-up of the OS to be resident in a memory. If each of these file names is described in advance in a box for storing file names of the first stored information 41 and data for checking tamper, for example, a digital signature is described, the tamper detecting unit 33 is allowed to check presence or absence of tamper of files which are executed or referred to at boot-up of the OS.

[0044] The tamper detecting function 3 can be adopted in detecting presence or absence of tamper of the first stored information 41 per se. The purpose for checking presence or absence of tamper of the first stored information 41 per se is as follows. For example, if one file to be checked 42 is tampered, description added to the file to be checked 42 may be deleted from the first stored information 41 to avoid tamper from being checked so as to conceal the tamper. Check of the first stored information is useful for detecting presence or absence of concealment of the tamper. Specifically, as shown in Fig. 3 for example, tamper check stored information on

information obtained up to that time, for example a digital signature, is allocated to the end of the first stored information 41, and then, the tamper detecting unit 33 has only to utilize the tamper check stored information and the second stored information in the boot program 2 to check presence or absence of the tamper.

[2] Descriptions of Operation

Then, an operation of the embodiment of the present invention is described with reference to Figs. 1 and 2.

[0045] When a computer system is turned on, an execution address is set as a leading address of a boot program 2 stored in a ROM 1 so as to execute the boot program 2.

[0046] Fig. 2 is a flowchart that shows a processing flow of the boot program 2. The boot program 2 first starts processing of initializing periphery devices (step A1). This step is for execution of test and initialization of connected periphery devices and initialization of a BIOS, thereby to make the BIOS usable. The steps A2 to A10 are features of the present invention.

[0047] At first, at step A2, the above-mentioned stored information reading unit 31 is used to read the first stored information 41.

[0048] Then, at step A3, the first stored information 41 is subjected to tamper check. When this is performed using a digital signature, first, a portion other than the own signature data of the first stored information 41 stored at the end of the first stored information 41 as shown in Fig. 3 is compressed by Hash function. Then, the own signature data of the first stored information 41 stored at the end thereof is decoded by a key stored in the second stored information 34 in the boot program 2. This compressed value and the decoded value are compared to check presence or absence of tamper depending on coincidence of the comparison result with the

decoded value. If they are coincident, there has occurred no tampering and the processing proceeds to step A5. When they are not coincident, the processing proceed to step A8 (step A4).

[0049] Step A5 is a step of confirming that tamper check is performed for each file to be checked 42 written in the first stored information 41. When all of the files to be checked 42 are not checked, the processing proceeds to step A6. When all the files to be checked are completed, the processing proceeds to step A11.

[0050] At step A6, the above-mentioned file to be checked reading unit 32 is used to read a file to be checked 42.

[0051] At step A7, the file to be checked 42 is subjected to tamper check. When tamper check is performed by using a digital signature, first, the file to be checked is compressed by Hash function. Then, a signature data stored in the first stored information 41 is decoded by a key which is stored in the second stored information 34 in the boot program 2. The resultant compressed value and the decoded value are compared to check presence or absence of tamper depending on coincidence of the comparison result with the decoded value. If they are coincident, there has occurred no tampering and the processing proceeds to step A5. When they are not coincident, the processing proceeds to step A8 (step A4).

[0052] At step A4, when tampering is detected, presence of tampering is displayed according to need in an additionally connected display, and it is determined whether the boot-up operation is continued or not (step A8, step A9). When continuation command is input at an additionally connected input device, it proceeds to step A5. When discontinue command is input, the boot program processing is terminated to abort the boot-up processing (step A10).

[0053] When at step A5, all the files to be checked are completely checked,

execution processing of the OS loader 43 is performed at step A11. The OS loader 43 written in a boot sector of the auxiliary storage 4 is read in the RAM 102 by the BIOS, an execution address of the CPU is set at the read RAM address, thereby to transfer the control to the OS loader 43. Then, the boot program processing is finished.

[3] Other Embodiments of the Invention

Next, a second embodiment of the present invention will be described with reference to Fig. 4.

[Effects of the Invention] The first effect is in that presence or absence of tamper of an OS loader, an OS file or any other files can be detected with reliability and without users' consciousness before boot-up of the OS loader or OS. In other words, it is a boot program that performs the above-mentioned detecting, which is executed with reliability at boot-up of the computer system. Since the boot program is stored in the ROM, there is no possibility of the program per se being tampered.

[0093] A second effect is in the simplicity of constituting a hardware configuration. In other words, the present invention can be configured by adding a soft module for performing tamper check to a boot ROM generally mounted in a system and by arranging files to be checked in physically sequential positions in an auxiliary storage 4 and using a utility program for generating first stored information 41, and any special hardware does not have to be added.

[Brief Description of the Drawings]

[Fig. 1] A view that show a configuration of an embodiment of the present invention.

[Fig. 2] A flowchart that explains an operation of the present invention.

[Fig. 3] A view that explains an example of first stored information.

[Fig. 12] A view that explains the conventional art.

[Description of the Symbols]

1	ROM
2	Boot program
3	Tamper detecting function
31	Stored information reading unit
32	File to be checked reading unit
33	Tamper detecting unit
34	Second stored information
35	Stored information inputting unit
36	Second stored information reading unit
37	Communicating unit
38	Tamper detecting program executing unit
39	Tamper detecting program checking unit
391	Tamper detecting program compressed value
4	Auxiliary storage
41	First stored information
42	File to be checked
43	OS loader
44	OS file
45	Boot sector
46	Alternate OS loader
461	Tamper detecting program executing unit
462	OS loader executing unit
47	Tamper detecting program

	48	File to be checked list
	6	Input device
	*7	External storage
	71	Anti-tamper unit
5	8	External computer
	81	Anti-tamper unit
	82	Tamper detection calculating unit
	9	Tamper detecting program
	91	Write protect function storage
10	100	Controller
	101	CPU
	102	RAM

Fig. 1

15	100	Controller
	2	Boot program
	3	Tamper detecting function
	31	Stored information reading unit
	32	File to be checked reading unit
20	33	Tamper detecting unit
	34	Second stored information
	4	Auxiliary storage
	41	First stored information
	42	File to be checked
25	43	OS loader

44 OS file

Fig. 3

Leading marker

Whole size

Number of files to be checked (n)

First checked file name

Signature data of first checked file

Second checked file name

Signature data of second checked file

nth checked file name

Signature data of nth checked file

Own signature data of first stored information

Fig. 12

A

In preparation

Program file

Signed data

Signature generating process

Compressing by Hash function

Compressed value

Code by confidential key

Confidential key

B

In checking

Program file

Signed data

Signature verifying processing

Compressing by Hash function

Compressed value

Compare

Decoded value

Decode by public key

Public key

(19)日本国特許庁（ＪＰ）　(12) **公開特許公報**（Ａ）　(11)特許出願公開番号

特開平10－333902

(43)公開日　平成10年(1998)12月18日

(51)Int.Cl.[6]		識別記号	FI		
Ｇ０６Ｆ	9/06	５５０	Ｇ０６Ｆ	9/06	５５０Ｚ
	9/445			12/14	３１０Ｚ
	12/14	３１０		9/06	４２０Ｇ

審査請求　有　　請求項の数13　ＯＬ　（全 14 頁）

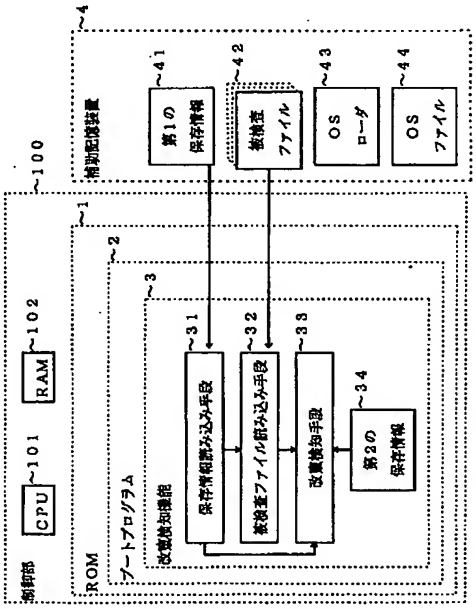
(21)出願番号　特願平9－137181

(22)出願日　平成９年(1997)５月27日

(71)出願人　394017491
株式会社エヌイーシー情報システムズ
神奈川県川崎市高津区坂戸３丁目２番１号
(72)発明者　望月　和広
神奈川県川崎市高津区坂戸３－２－１　株式会社エヌイーシー情報システムズ内
(74)代理人　弁理士　京本　直樹　（外２名）

(54)【発明の名称】　改ざん検知機能付きコンピュータシステム

(57)【要約】
【課題】コンピュータの起動時に、ＯＳ、ＯＳローダやその他の被検査ファイルの改ざんの有無を検査し、安全確認後にＯＳ、ＯＳローダ、アプリケーションを起動する。
【解決手段】ＲＯＭ１に格納されたブートプログラム２は、ＢＩＯＳの初期化の後に、補助記憶装置４に格納された第１の保存情報４１をＯＳを介さずに読み込み、その読み込み内容にしたがって被検査ファイル４２をＯＳを介さずに読み込む。これらの読み込み内容とＲＯＭ１に格納された第２の保存情報とを用いて被検査ファイルの有無を検査する。改ざん無しを確認後に、ＯＳローダ４３を呼出してＯＳ４４を起動させる。



【特許請求の範囲】

【請求項１】起動するとまずＲＯＭ内のブートプログラムが起動し、

前記ブートプログラムは補助記憶装置内のＯＳローダを起動し、

前記ＯＳローダは前記補助記憶装置内のＯＳを起動するコンピュータシステムにおいて、

前記ブートプログラムが、

前記補助記憶装置に格納された第１の保存情報を読み込む保存情報読み込み手段と、

読み込んだ前記第１の保存情報の記載内容を元に前記補助記憶装置から任意個の被検査ファイルを読み込む被検査ファイル読み込み手段と、

前記ブートプログラムの内部に持つ第２の保存情報と読み込んだ前記第１の保存情報と前記各被検査ファイルとから前記各被検査ファイル個々の改ざんの有無を検査する改ざん検知手段と、を備えた改ざん検査機能を持つことを特徴としたコンピュータシステム。

【請求項２】前記第１の保存情報に前記各被検査ファイルに対するデジタル署名を格納し、

前記第２の保存情報に署名検証用の鍵を格納し、

請求項１に記載の改ざん検知手段が前記デジタル署名と前記署名検証用の鍵とで各被検査ファイルの改竄の有無を検査することを特徴としたコンピュータシステム。

【請求項３】前記補助記憶装置に格納されたＯＳローダを、前記被検査ファイルの１つとして、ブート時に改ざんの有無の検査することを特徴としたコンピュータシステム。

【請求項４】前記補助記憶装置に格納されたＯＳファイルを、前記被検査ファイルの１つとして、ブート時に改ざんの有無の検査することを特徴としたコンピュータシステム。

【請求項５】前記第１の保存情報内に、自分自身の改ざん検査用の情報を持ち、

請求項１に記載の改ざん検知手段が、前記第１の保存情報の改ざんの有無を検査する機能をも有することを特徴としたコンピュータシステム。

【請求項６】前記第１の保存情報に前記各被検査ファイルを一方向性関数で圧縮した圧縮値と、前記第１の保存情報に対するデジタル署名とを格納し、

前記第２の保存情報に署名検証用の鍵を格納し、

前記改ざん検知手段は、前記デジタル署名と前記署名検証用の鍵とで第１の保存情報の改ざんの有無を検査し、前記圧縮値で各被検査ファイルの改ざんの有無を検査することを特徴としたコンピュータシステム。

【請求項７】前記ブートプログラムは、入力装置から前記第２の保存情報を取得する保存情報取得手段を備え、前記第２の保存情報を使用者の入力から取得することを特徴とするコンピュータシステム。

【請求項８】請求項７において、

前記第１の保存情報に、前記各被検査ファイルと前記第２の保存情報とを一方向性関数で圧縮した圧縮値と、

前記第１の保存情報自身と前記第２の保存情報とを一方向性関数で圧縮した圧縮値と、を予め格納しておき、

前記改ざん検知手段は、請求項７に記載の前記保存情報取得手段で取得した前記第２の保存情報を用いて再度前記圧縮値を計算し、前記第１の保存情報に格納された前記圧縮値と比較することで、前記各被検査ファイルと前記第１の保存情報の改ざんの有無を検査することを特徴としたコンピュータシステム。

【請求項９】前記第２の保存情報を前記外部記憶装置に持ち、

前記ブートプログラムは前記外部記憶装置から第２の保存情報を取得する第２の保存情報読み込み手段を備え、

前記外部記憶装置は前記第２の保存情報の改ざん防止手段を備えることを特徴としたコンピュータシステム。

【請求項１０】前記第２の保存情報を外部コンピュータに持ち、

前記外部コンピュータは、

前記第２の保存情報の改ざん防止手段と、

前記第２の保存情報を用いて改ざん検知に必要な計算を行なう改ざん検知計算手段とを備え、

前記ブートプログラムは、前記外部コンピュータとの通信手段を備え、

請求項１に記載の前記改ざん検知手段は、前記第１の保存情報と、前記各被検査ファイルと、前記外部コンピュータとの通信手段を介して前記外部コンピュータ側の改ざん検知計算手段を使用することとで、前記各被検査ファイル個々の改ざんの有無を検査することを特徴としたコンピュータシステム。

【請求項１１】請求項１に記載の改ざん検知機能を持つ改ざん検知プログラムをライトプロテクト機能付きの記憶装置に格納し、

前記ブートプログラムは、前記改ざん検知プログラムの実行手段を備えることを特徴とするコンピュータシステム。

【請求項１２】請求項１に記載の改ざん検知機能を持つ改ざん検知プログラムを前記補助記憶装置に格納し、

前記改ざん検知プログラムを一方向性関数で圧縮した値を、予め前記ＲＯＭ内に保存しておき、

前記ブートプログラムは、前記改ざん検知プログラムを一方向性関数で圧縮し、その値を保存しておいた前記圧縮値と比較することにより、前記改ざん検知プログラム自身の改ざんの有無を検査する改ざん検知プログラム検査手段と、

前記改ざん検知プログラムを実行する改ざん検知プログラム実行手段と、を備えることを特徴とするコンピュータシステム。

【請求項１３】請求項１に記載の改ざん検知機能を備えた改ざん検知プログラムを前記補助記憶装置に格納し、

さらに前記補助記憶装置のブートセクタに格納されている前記ＯＳローダを予め前記補助記憶装置のブートセクタ以外の場所に保存しておき、
前記改ざん検知プログラムを前記補助記憶装置から読み込み実行する改ざん検知プログラム実行手段と、
保存された前記ＯＳローダを読み込み実行するＯＳローダ実行手段と、を備えた代替ＯＳローダを前記補助記憶装置のブートセクタに持たせることを特徴とするコンピュータシステム。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、改ざん検知機能付きコンピュータシステムに関し、特にコンピュータシステムの起動時にファイルの改ざんの検知を行なう改ざん検知機能付きコンピュータシステムに関する。

【０００２】

【従来の技術】従来、この種の改ざん検知機能付きコンピュータシステムは、例えば第１の例として「インテグリティチェック法を中心としたコンピュータウイルス対策システムの研究報告書」（１９９６年３月　情報処振興事業協会発行）に示されるように、コンピュータウイルス（以下単にウイルスと呼ぶ）によるプログラムファイルの改ざんを検知するために用いられている。この報告書に示されているシステムはデジタル署名と呼ばれる暗号技術を用いてウイルスによるプログラムファイルの改ざんを検知している。この基本原理を図１２を用いて説明する。

【０００３】改ざんの有無を検査すべきプログラムファイルには予めデジタル署名（以下単に署名と呼ぶ）を生成しておく。これは署名生成プログラムで、まず検査すべきプログラムファイルをＨａｓｈ関数と呼ばれる関数で圧縮し、圧縮値を秘密の鍵（以下秘密鍵と呼ぶ）で暗号化し、その暗号データを署名とする（図１２Ａ）。

【０００４】検査時には、プログラムファイルを再度Ｈａｓｈ関数で圧縮した値と、署名を公開された鍵（以下公開鍵と呼ぶ）で復号化した値とを比較する（図１２Ｂ）。プログラムが改ざんされていなければこの２つの値は一致する。一致しなければ何らかの改ざんがあったと判断できる。

【０００５】ここで用いる暗号方式は、非対称暗号あるいは公開鍵暗号と呼ばれるもので、暗号化する鍵と復号化する鍵が異なり、一方から他方が推測できない性質を持つものである。プログラムファイルの改ざんを隠蔽するために署名を偽造することは秘密鍵を知り得ない限り不可能である。

【０００６】またＨａｓｈ関数とは入力データを１６０ビット程度に圧縮する一方向性関数である。一方向性関数とは、ある値に変換される元の値を逆算することが計算量的に不可能な関数を言う。従って改ざんを隠蔽するためにＨａｓｈ関数による圧縮値が変らないように改ざん内容を調整する必要があるが、これも不可能である。

【０００７】前述の報告書付録５ページには、この従来技術の試作システム「ＩＰＡＩｎＣＳ」の運用上の注意が記述されている。そこには、ＯＳと署名検証システムをライトプロテクトを施したＦＤで保持し、そのＦＤでコンピュータを立ち上げてから実行する方法が安全だとしている。これはＯＳローダやＯＳファイル、もしくは署名検証システム自身がウイルスに感染した場合に、署名検証システムで改ざんの有無を検査をする以前にウイルスが動作してしまうからである。このウイルスがステルス機能を持つものであれば、後から感染（つまり改ざん）されたプログラムファイルを検査しても、改ざんは隠蔽されて検知できない。ステルス機能が検査のためにプログラムファイルを読み込むのを事前に察知し、改ざん部分を修復した読み込み結果を渡すからである。

【０００８】我が国におけるコンピュータウィルス感染被害の報告の半数以上が、ＯＳローダに感染するブートセクタ感染ウイルスである現在、前記のような運用方法をとる必要性は高い。

【０００９】プログラムファイルの改ざんを防止するシステムの例としては、特開平６－２３０９５９号公報に示されるものがある。このシステムでは、専用の拡張ボードを用いて、機能を拡張したＢＩＯＳとＤＯＳを動作させることにより、プログラムファイルが不正に書き換えられることを防止している。しかしこのシステムは、万が一プログラムファイルの改ざんがあった後にそれを検知できる仕組みではなく、本発明とは目的が異なるので、本方式の代替の方法となるものではない。しかも、ウイルスが感染のためにプログラムファイルを改ざんするのを監視するものであり、任意のファイル（例えばセキュリティの設定や暗号鍵などを格納したデータファイル）の改ざんを防止するものではない。さらに、不正にはアクセスできないＲＯＭとＲＡＭを持つハードウエア構成を必須とする装置構成上の制約がある。

【００１０】

【発明が解決しようとする課題】従来技術の問題点は、操作性の問題である。

【００１１】その理由は、前記従来の技術の第１の例で示したように、安全に運用するためには、物理的に書き換え不能な安全な媒体でコンピュータシステムを立ち上げて検査を実行しなければならない。しかし実際問題としては、ほとんどのコンピュータ使用者は通常、ＯＳやアプリケーションプログラムはハードディスクに保持している。ハードディスクは物理的に書き換え不能な媒体ではなく、改ざんの危険性が高い。従ってＯＳと改ざん検査プログラムの格納されたライトプロテクトを施したフロッピーディスクで立ち上げ、ハードディスク内のＯＳやＯＳローダを検査した後に、再度ハードディスクで立ち上げると言った方法を取らざるを得ない。これは使用者に操作の煩わしさと、余分なシステムの立ち上げ待

ち時間を強いることになる。

【００１２】本発明の主たる目的は、使用者にストレスを感じさせない操作性のもとで、ＯＳローダやＯＳや他のコンピュータ起動時に実行されるプログラムの改ざんの有無を確実に検査することである。

【００１３】本発明の他の目的は、第１の目的を実現するための装置構成を簡素化する点である。

【００１４】本発明の他の目的は、改ざん検知処理の高速化を図る点である。

【００１５】本発明の他の目的は、システム全体の保守性を向上させる点である。

【００１６】

【課題を解決するための手段】本発明の第１の実施形態では、ＲＯＭ１に格納されたブートプログラム２に、補助記憶装置４に格納された任意個の任意の被検査ファイルの改ざんの有無の検査する機能を持たせる。より具体的には、補助記憶装置４に格納された第１の保存情報４１を読み込む保存情報読み込み手段（図１の３１）と、読み込んだ第１の保存情報４１の記載内容を元に補助記憶装置４から任意個の被検査ファイル４２を読み込む被検査ファイル読み込み手段（図１の３２）と、ブートプログラム２の内部に持つ第２の保存情報３４と読み込んだ第１の保存情報４１と各被検査ファイル４２とから各被検査ファイル個々の改ざんの有無を検査する改ざん検知手段（図１の３３）とを有する。

【００１７】本発明の第２の実施形態では、第１の実施形態で記載した第２の保存情報３４を使用者の入力から取得することを特徴とする。より具体的には、ブートプログラム２が入力装置６から第２の保存情報３４を取得する保存情報取得手段（図５の３５）を有する。

【００１８】本発明の第３の実施形態では、第２の実施形態において一方向性関数による圧縮値で改ざんの有無を検査する特徴を持つ。より具体的には、入力装置６から第２の保存情報３４を取得する保存情報取得手段（図５の３５）と、被検査ファイル４２と第２の保存情報３４とを一方向性関数で圧縮した圧縮値と予め第１の保存情報４１に保存された圧縮値とを比較して改ざんの有無を判定する改ざん検知手段（図５の３３）とを有する。

【００１９】本発明の第４の実施形態では、第２の保存情報３４を外部記憶装置７に持つことを特徴とする。より具体的には、ブートプログラム２は外部記憶装置７から第２の保存情報３４を取得する第２の保存情報読み込み手段（図７の３６）を有し、外部記憶装置７は第２の保存情報３４の改ざん防止手段（図７の７１）を有する。

【００２０】本発明の第５の実施形態では、第２の保存情報３４を外部コンピュータ８に持ち、改ざん検知に必要な計算処理を外部コンピュータ８側で行なわせる。より具体的には、外部コンピュータ８は、第２の保存情報３４の改ざん防止手段（図８の８１）と、改ざん検知に必要な計算を行なう改ざん検知計算手段（図８の８２）とを備え、ブートプログラム２は、外部コンピュータ８との通信手段（図８の３７）と、通信手段３７を介して外部コンピュータ８側の改ざん検知計算手段８２を使用して各被検査ファイル個々の改ざんの有無を検査する改ざん検知手段（図８の３３）を有する。

【００２１】本発明の第６の実施形態では、改ざん検知機能３を改ざん検知プログラム９をライトプロテクト機能付きの記憶装置９１に保持してブート時に実行する。より具体的には、ブートプログラム２は、改ざん検知プログラム９の実行手段（図９の３８）を有する。

【００２２】本発明の第７の実施形態では、改ざん検知機能３を改ざん検知プログラム９を補助記憶装置４にに保持してブート時に検査して実行する。より具体的には、ブートプログラム２は、一方向性関数を用いて改ざん検知プログラム９の改ざんの有無を検査する改ざん検知プログラム検査手段（図１０の３９）と、改ざん検知プログラム９を実行する改ざん検知プログラム実行手段（図１０の３８）とを有する。

【００２３】本発明の第８の実施形態では、ＯＳローダ４３に改ざん検査機能３を持たせる。より具体的には、代替ＯＳローダ４６が、改ざん検知機能３を備えた改ざん検知プログラム９を前記補助記憶装置４から読み込み実行する改ざん検知プログラム実行手段（図１１の４６１）と、別途保存されたＯＳローダ４３を読み込み実行するＯＳローダ実行手段（図１１の４６２）とを有する。

【００２４】

【作用】本発明の第１の実施形態では、保存情報読み込み手段は、被検査ファイル読み込み手段に対して、被検査ファイルを読み込むための情報、具体的には補助記憶装置内に被検査ファイルが格納されている物理的位置の情報を提供する。また、改ざん検知手段に対しては、被検査ファイルの改ざんの有無を判定するための情報、具体的には被検査ファイルに対する署名データなどを提供する。

【００２５】被検査ファイル読み込み手段は、改ざん検知手段に対して、現在の被検査ファイルの内容を提供する。

【００２６】改ざん検知手段は、提供された被検査ファイルの改ざんの有無を判定するための情報と、現在の被検査ファイルの内容と、ブートプログラム２の内部に持つ第２の保存情報（具体的には署名検証用の鍵など）から、被検査ファイルの改ざんの有無を検査する。

【００２７】本発明の第２の実施形態と第３の実施形態では、保存情報取得手段が、改ざん検知手段に、第２の保存情報を提供する。

【００２８】本発明の第４の実施形態では、外部記憶装置に保存された第２の保存情報は、外部記憶装置の改ざん防止手段で改ざんから守られる。保存情報読み込み手

段は第２の保存情報を外部記憶装置から取得し、改ざん検知手段に提供する。

【００２９】本発明の第５の実施形態では、外部コンピュータに保存された第２の保存情報は、外部外部コンピュータの改ざん防止手段で改ざんから守られる。ブートプログラムは周辺機器の初期化処理後、改ざん検知手段で、通信手段を介して外部コンピュータとデータの送受信を行ない、外部コンピュータ側の改ざん検知計算手段を利用して、各被検査ファイルの改ざんの有無を検査する。

【００３０】本発明の第６の実施形態では、ブートプログラムは周辺機器の初期化処理後、改ざん検知プログラム実行手段で、ライトプロテクト機能付きの記憶装置に格納された改ざん検知プログラムを実行する。

【００３１】本発明の第７の実施形態では、ブートプログラムは周辺機器の初期化処理後、改ざん検知プログラム検査手段で、補助記憶装置４に格納された改ざん検知プログラムに改ざんがないかチェックする。次いで、改ざん検知プログラム実行手段で改ざん検知プログラムを実行する。これで被検査ファイルを検査する。

【００３２】本発明の第８の実施形態では、ブートプログラムにより代替ＯＳローダが実行され、代替ＯＳローダはまず、改ざん検知プログラム実行手段で改ざん検査機能を持った改ざん検知プログラムを実行する。これでこれから起動するＯＳに必要な被検査ファイルを検証する。次いで、代替ＯＳローダ４６は、ＯＳローダ実行手段で、別途保存されたＯＳローダ４３を読み込み実行し、ＯＳが起動する。

【００３３】

【発明の実施の形態】

［１］構成の説明

次に、本発明の実施の形態について図面を参照して詳細に説明する。

【００３４】図１を参照すると、本発明の第１の実施の形態は、制御部１００と、補助記憶装置４とを含む。

【００３５】制御部１００は、ＲＯＭ１とＲＡＭ１０２と、それらに格納されたプログラムを実行するＣＰＵ１０１とを含む。

【００３６】ＲＯＭ１に格納されたブートプログラム２は、補助記憶装置４を含む周辺機器の初期設定を行ない基本入出力システム（以下ＢＩＯＳと略す）を使用可能にした後、補助記憶装置４に格納されたＯＳローダ４３をＲＡＭ１０２に読み込み実行するプログラムである。本発明は、このブートプログラム２に、改ざん検知機能３を持たせることを特徴とする。

【００３７】補助記憶装置４には、ＯＳファイル４４と、ＯＳファイル４４をＲＡＭ１０２に読み込み実行するためのＯＳローダ４３ととが含まれている。本発明では、この補助記憶装置４に任意個の任意のファイルを、システム立ち上げ時に改ざんの有無を検査する被検査ファイル４２として設定でき、その設定内容と、改ざん検査用の情報、例えばディジタル署名などの情報を第１の保存情報４１に格納していることを特徴としている。

【００３８】前述の改ざん検知機能３は、保存情報読み込み手段３１と、被検査ファイル読み込み手段３２と、改ざん検知手段３３とを備える。

【００３９】保存情報読み込み手段３１は、補助記憶装置４から第１の保存情報４１を読み込むものである。この処理を行なう時点、つまりブートプログラム２が起動する時点は、まだＯＳの起動前であり、ＯＳを介したファイルアクセスは使用できない。従ってＢＩＯＳの低レベルな関数だけを使って補助記憶装置４をアクセスする必要がある。例えば、ＯＳがマイクロソフト社製のＭＳ－ＤＯＳであれば、補助記憶装置４のディレクトリデータとファイルアロケーションテーブル（以下ＦＡＴと略す）を参照して、各ファイルの格納された補助記憶装置４内での物理的位置の情報を得ることができる。つまりディレクトリデータにはファイル名とＦＡＴのエントリポイントの対応が記述されており、そのエントリポイントから順にＦＡＴを辿ることにより、ファイルが格納された物理的格納位置を知ることができる。従って予め第１の保存情報４１のファイル名を固定しておけば、そのファイル名から補助記憶装置４内の物理的格納位置を取得することができ、ＢＩＯＳを使って読み出せばよい。他のＯＳの場合も同様に、そのＯＳが行なうファイルの物理的格納位置の取得方法を実行し、ＢＩＯＳで読み出せばよい。

【００４０】被検査ファイル読み込み手段３２は、読み込んだ第１の保存情報４１から被検査ファイル４２の設定内容を取得し、その被検査ファイル４２を補助記憶装置４から読み込むものである。ここでもＢＩＯＳの低レベルな関数だけを使っての記憶装置にアクセスする必要がある。これも前述の保存情報読み込み手段３１と同様に、使用するＯＳが行なうのと同じ方法で補助記憶装置４内の物理的格納位置を取得し、ＢＩＯＳを使って読み出せばよい。

【００４１】改ざん検知手段３３は、補助記憶装置４から読み込んだ各被検査ファイルに対して、第１の保存情報４１から読み取った検査用の情報と、ブートプログラム自身が持つ第２の保存情報３４とから、改ざんの有無を検査する。例えば、第１の保存情報４１として、各被検査ファイルに対するディジタル署名を格納し、第２の保存情報３４として、署名検証用の鍵を格納しておく。改ざん検知手段３３は前記ディジタル署名と前記署名検証用の鍵とで各被検査ファイルの改ざんの有無を検査する。

【００４２】また、この改竄検知機能３で、ＯＳローダ４３の改竄の有無を検証することもできる。ＯＳローダは補助記憶装置４のブートセクタと呼ばれる部分に格納されている。ブートセクタは補助記憶装置の物理的先

頭、もしくは複数のパーティションが切られているときは各パーティションの先頭に位置する。第１の保存情報４１のファイル名の格納欄に検査したいブートセクタのドライブ名を記述しておけば、その補助記憶装置４内での物理的位置は特定できる。更にブートセクタに対する改ざん検査用のデータ、例えばデジタル署名を格納しておけば、この改ざん検知手段３３で、ブートセクタの改ざんの有無、つまりはその中に格納されたＯＳローダ４３の改ざんの有無を検査できる。厳密にはＯＳローダ４３はブートセクタの一部であるので、ブートセクタの改ざんの有無と、ＯＳローダ４３の改ざんの有無は異なるが、ＯＳローダ４３のプログラム部分以外のブートセクタには記録フォーマットなどの情報が格納されているので、ここも含めた改ざんの検査をすれば問題ない。プログラム部分だけに限定させるなら、例えば、第１の保存情報４１に先頭の読み飛ばしバイト数と終端の無視するバイト数を格納するフィールドを追加すれば、検査すべきプログラム部分のみを特定できる。

【００４３】改ざん検知機能３でＯＳファイル４４の改ざんの有無を検証することもできる。　ＯＳファイル４４とは、ここでは基本となるＯＳプログラムと、それが読み込む設定ファイルと、およびＯＳ起動時に実行される各種ドライバ類の総称とする。例えばマイクロソフト社のＭＳ－ＤＯＳを例に取ると、基本となるＯＳプログラムはファイル名が「ｉｏ．ｓｙｓ」、「ｍｓｄｏｓ．ｓｙｓ」、「ｃｏｍｍａｎｄ．ｃｏｍ」などのファイルで、それらが読み込む設定ファイルは「ｃｏｎｆｉｇ．ｓｙｓ」や「ａｕｔｏｅｘｅｃ．ｂａｔ」などのファイルで、ドライバ類は前記設定ファイルに記述されたものがＯＳ起動時に実行されメモリ中に常駐する。これらのファイル名を予め第１の保存情報４１のファイル名の格納欄に記述し、さらに改ざん検査用のデータ、例えばデジタル署名を記述しておけば、前記改ざん検知手段３３でＯＳの起動時に実行もしくは参照されるファイルの改ざんの有無を検査できる。

【００４４】改竄検知機能３で第１の保存情報４１自身の改ざんの有無を検証することもできる。この第１の保存情報４１自身の改ざんの有無を検査する目的は、次の通りである。例えば、ある被検査ファイル４２が改ざんされた際に、その被検査ファイル４２に対する記述を第１の保存情報４１から削除してしまい、検査を逃れて改ざんを隠蔽されることが有り得る。この第１の保存情報の検証は、この隠ぺいの事実の有無を検出するのに有益である。具体的な方法としては、例えば図３で示すように第１の保存情報４１の末尾に、そこまでの情報に関する改ざん検査用保存情報、例えばデジタル署名を付与しておき、改ざん検知手段３３は、この改ざん検査用保存情報とブートプログラム２内の第２の保存情報とを用いて改ざんの有無を検査すればよい。

［２］　動作の説明

次に、図１および図２を参照して、本発明の実施の形態の動作について説明する。

【００４５】コンピュータシステムの電源を入れると、ＣＰＵ１０１の実行アドレスはＲＯＭ１に格納されたブートプログラム２の処理の先頭アドレスにセットされ、ブートプログラム２の実行が開始される。

【００４６】図２はブートプログラム２の処理の流れを示すフローチャートである。ブートプログラム２はまず周辺機器の初期化の処理を実行する（ステップＡ１）。これは、接続された周辺機器のテストや初期設定の実行と、ＢＩＯＳの初期設定などを行なうものである。これによりＢＩＯＳが使用可能になる。ステップＡ２からステップＡ１０までが本発明の特徴となる部分である。

【００４７】まずステップＡ２で前述の保存情報読み込み手段３１を用いて、第１の保存情報４１を読み込む。

【００４８】次いでステップＡ３で第１の保存情報４１自身の改ざんの検査を行なう。これは例えばデジタル署名を使用する方法であれば、まず図３で示すように第１の保存情報４１の末尾に格納された第１の保存情報４１自身の署名データを除いた部分をＨａｓｈ関数で圧縮する。次いで末尾にある第１の保存情報４１自身の署名データを、ブートプログラム２内の第２の保存情報３４に格納された鍵を使って復号する。この圧縮値と復号値と比較して、一致するか否かで改ざんの有無を検査する。一致していれば改ざんは無いので、ステップＡ５に進み、一致していなければステップＡ８に進む（ステップＡ４）。

【００４９】ステップＡ５は、第１の保存情報４１に書かれた被検査ファイル４２の数だけ改ざんの有無を検査したどうかを確認するためのステップである。全部の被検査ファイル４２の検査が終えていない場合、ステップＡ６へと進み、全部の検査が終えてた場合はステップＡ１１へと進む。

【００５０】ステップＡ６は前述の被検査ファイル読み込み手段３２を用いて、各被検査ファイル４２を読み込む。

【００５１】次いでステップＡ７で、被検査ファイル４２の改ざんの検査を行なう。これは例えばデジタル署名を使用する方法であれば、まず被検査ファイルをＨａｓｈ関数で圧縮する。次いで第１の保存情報４１に格納された署名データを、ブートプログラム２内の第２の保存情報３４に格納された鍵を使って復号する。この圧縮値と復号値と比較して、一致するか否かで改ざんの有無を検査する。一致していれば改ざんは無いので、ステップＡ５に進み、一致していなければステップＡ８に進む（ステップＡ４）。

【００５２】ステップＡ４において、改ざんが検知された場合には、必要に応じて別途接続された表示装置に表示し、継続して起動処理を継続するか確認する（ステップＡ８、ステップＡ９）。別途接続された入力装置から

継続を指示された場合はステップＡ５へと進み、中止が指示された場合はブートプログラムの処理を終了させ、起動処理を中止する（ステップＡ１０）。

【００５３】ステップＡ５で全ての被検査ファイルを検査を終了した場合、ステップＡ１１でＯＳローダ４３の実行処理を行なう。これは補助記憶装置４のブートセクタに書かれたＯＳローダ４３をＢＩＯＳを使ってＲＡＭ１０２に読み込み、読み込まれたＲＡＭアドレスにＣＰＵの実行アドレスをセットすることにより、ＯＳローダ４３に制御を移す。これでブートプログラムの処理は終了する。

［３］発明の他の実施形態

次に本発明の第２の実施形態について図４を参照して説明する。

【００５４】図４は、第２の実施形態における、第１の保存情報４１の構成を示したものである。

【００５５】この実施形態では、第１の保存情報４１に各被検査ファイル４２を一方向性関数で圧縮した圧縮値と、第１の保存情報４１自身に対するデジタル署名とを格納しておく。このデジタル署名の検証用の鍵は、第１の実施形態と同じくブートプログラム２内の第２の保存情報３４に格納しておく。

【００５６】この実施形態におけるブートプログラム２の改ざん検知手段３３は、第１の実施形態と同じく、まず第１の保存情報４１自身に対する署名データで、第１の保存情報４１自身に改ざんが無いことを確認する。次いで被検査ファイル読み込み手段３２で読み込んだ被検査プログラムを一方向性関数で圧縮し、その圧縮値と第１の保存情報４１に格納された圧縮値とを比較する。被検査ファイル４２に改ざんが無い場合はこの値は一致する。被検査ファイル４２に改ざんがあった場合、現在の内容を圧縮した値と保存された圧縮した値は異なるので、改ざんを検知できる。

【００５７】上記の圧縮値は使用している一方向性関数の仕様さえ判れば生成することができる。従って被検査プログラム４２が改ざんされ、その改ざんを隠蔽するため、第１の保存情報４１内に保存された圧縮値までも、改ざん結果の圧縮値に書き換えられてしまうこともありうる。この場合、前述の圧縮値同士の比較は一致し、検査をパスされてしまうが、その前に第１の保存情報４１自身の改ざん検査で、改ざんの隠蔽があったことを検知できる。

【００５８】この実施形態においては、前述のように第１の保存情報４１自身までもが改ざんされ、被検査ファイル４２に改ざんが隠蔽された場合に、被検査ファイル４２のいずれかが改ざんされそれが隠蔽されたと検知できるものの、被検査ファイル４２のどれが改ざんされたのかの特定がこの方法だけではできないというデメリットはある。ただし署名復号の処理が１回で済むため全ての被検査ファイル４２を検証する処理時間は短時間で済む。従って処理速度を重視する場合には有用である。

【００５９】次に本発明の第３の実施形態について図５を参照して説明する。

【００６０】図５は、ブートプログラム２の改ざん検知機能３の一部として、入力装置６を用いて前述の第２の保存情報３４を使用者の入力操作により取得し、改ざん検知手段３３で用いることを特長としている。

【００６１】ブートプログラム２は、図２のステップＡ１の周辺機器の初期化の後で、入力装置６から第２の保存情報３４を取得する。これはステップＡ１で入力装置６の初期化とＢＩＯＳの初期化が完了しているので、ＢＩＯＳを使って入力装置６から入力情報を受け取ればよい。

【００６２】この実施形態は、第２の保存情報３４を使用者の入力操作により取得するため、大きなサイズの情報を使用できないというデメリットがあるものの、第２の保存情報３４をブートプログラム２の格納されたＲＯＭ１から切り離すことで、情報を更新できるという効果がある。さらに第２の保存情報３４をＲＯＭ１から切り離すことにより、改ざんを企てる第３者から秘匿することができる。その結果次の第４の実施形態が可能になる。

【００６３】次に本発明の第４の実施形態について図６を参照して説明する。

【００６４】図６は、前記第４の実施形態における、第１の保存情報４２の例である。

【００６５】この例では、第１の保存情報４１に、各被検査ファイル４２と使用者だけが知り得る第２の保存情報とを連結し、一方向性関数で圧縮した圧縮値を格納する。ここでいう連結とは、各被検査ファイル４２の末尾に第２の保存情報を繋げることを言う。連結する以外にも、２つの情報の排他的論理和（ＸＯＲ）を取る方法でも構わない。第１の保存情報４１自身も、第２の保存情報３４と連結して一方向性関数で圧縮した圧縮値を、自分自身の末尾に持たせておく。

【００６６】改ざん検知手段は、検査時に、前記保存情報取得手段で取得した前記第２の保存情報とを連結し、圧縮値を計算し、第１の保存情報４１に格納された値と比較することで、前記各被検査ファイルの改ざんの有無を確認する。第１の保存情報４１自身の改ざんの有無も同様に確認する。

【００６７】被検査プログラムの改ざんを隠蔽するために、この保存情報を偽造することは、使用者だけが知り得る第２の保存情報を知り得ない限り、一方向性関数の性質上、計算量的に困難である。

【００６８】この実施形態は、前記第３の実施形態の効果に加え、暗号処理を含まないため処理の高速化と、ブートプログラム２が簡素化できる効果がある。

【００６９】次に本発明の第５の実施形態について図７を参照して説明する。

【００７０】この実施形態では、第２の保存情報３４を外部記憶装置７に持ち、ブートプログラム２は外部記憶装置７から第２の保存情報を取得する第２の保存情報読み込み手段３６を備え、なおかつ外部記憶装置７は第２の保存情報３４の改ざん防止手段７１を備えることを特徴としている。

【００７１】外部記憶装置７は例えば、フロッピーディスクとそのドライブで構成することができる。この場合は、第２の保存情報読み込み手段３６は、ＢＩＯＳを用いでドライブをアクセスすることで第２の保存情報３４を取得できる。改ざん防止手段７１は、フロッピーディスクにライトプロテクトを施せばよい。

【００７２】この実施形態は、第２の保存情報３４をブートプログラム２の格納されたＲＯＭ１から切り離すことで、保存情報の更新が容易にできるという効果がある。

【００７３】次に本発明の第６の実施形態について図８を参照して説明する。

【００７４】この実施例では、第２の保存情報３４を外部コンピュータ８に持ち、この外部コンピュータは、第２の保存情報の改ざん防止手段８１と、第２の保存情報を用いて改ざん検知に必要な計算を行なう改ざん検知計算手段８２とを備える。さらにブートプログラム２は、外部コンピュータ８との通信手段３７を備える。そして、ブートプログラム２の改ざん検知手段３３は、第１の保存情報４１と各被検査ファイル４２と、外部コンピュータとの通信手段３７を介して外部コンピュータ８側の改ざん検知計算手段８２を使用することとで、各被検査ファイル４２個々の改ざんの有無を検査するものとする。

【００７５】この実施形態でいう外部コンピュータ８は、例えば、公開鍵暗号の計算機能を備えたＩＣカードとそのリーダ／ライタとで構成できる。

【００７６】公開鍵暗号の計算機能を備えたＩＣカードとしては、ＣＰ８トランサック（Ｔｒａｎｓａｃ）社製のＴＢ９８Ｓなどが知られている。このようなカードを用いれば、前述の改ざん検知計算手段８２として、署名の復号化処理をブートプログラム２から切り離し、ＩＣカード側で行なうことができる。この場合ブートプログラム２側の改ざん検知手段３３は、被検査ファイル４２の署名をＩＣカードに送り、その復号値を受け取り、その値とブートプログラム２側で圧縮した値とを比較して改ざんを検査すればよい。

【００７７】外部コンピュータ２にＩＣカードを用いた場合の第２の保存情報の改ざん防止手段８１は、所定の手続き、例えば使用者のＰＩＮコードを照合しない限りデータの更新を許可しない機能で実現できる。この機能はＩＣカードが一般的に持つ機能である。

【００７８】また、外部コンピュータ２にＩＣカードを用いた場合のブートプログラム２側の通信手段３７は、例えばＲＳ２３２－Ｃポートを介して接続されるリーダ／ライタを使えば、実現できる。これは、ブートプログラム２は最初に周辺機器を初期化する（図２ステップＡ１）ので、検査を開始する時点ではＲＳ２３２－Ｃ関連のＢＩＯＳも使用可能であり、これを用いて通信を行なえばよい。

【００７９】この実施形態においては、改ざん検知に関する計算処理の一部をＲＯＭ１内のブートプログラムから切り離すことで、計算処理の変更やバージョンアップなどに対応しやすくなる効果がある。

【００８０】次に本発明の第７の実施形態について図９を参照して説明する。

【００８１】この実施形態では、改ざん検知機能３を持つ改ざん検知プログラム９をライトプロテクト機能付きの記憶装置９１に格納する。

【００８２】ライトプロテクト機能付きの記憶装置９１は、拡張ボードとそれに搭載された拡張ＲＯＭとで構成することができる。多くの拡張ボードには、そのボードを扱うためのＢＩＯＳが搭載された拡張ＲＯＭをボード上に持つ。これに対応すべく、多くのコンピュータシステムのブートプログラムは、ＢＩＯＳが格納された拡張ＲＯＭが装着されていないか拡張ＲＯＭ用のアドレス空間を検索し、見つけた場合はその初期ルーチンを呼び出す機能を持つ。改ざん検知プログラム９を拡張ボード上の拡張ＲＯＭに格納すれば、前述の初期ルーチンを呼び出す機能がそのまま、改ざん検知プログラム実行手段３８となる。

【００８３】ライトプロテクト機能付きの記憶装置９１は、他にフロッピーディスクとそのドライブ装置とでも構成できる。つまり、ブートプログラム２が記憶装置９１から改ざん検知プログラム９を読み込む方法は、例えば改ざん検知プログラム９を記憶装置９１の所定の物理的位置から始まる物理的連続領域に格納しておき、その所定の位置からＢＩＯＳを使って読み出せばよい。実行する方法は、ジャンプ命令でＣＰＵ１００のＩＰアドレスを、ＲＡＭ１０２に読み込まれた改ざん検知プログラム９の先頭アドレスに設定すればよい。

【００８４】この実施形態においては、改ざん検知機能３をＲＯＭ１内のブートプログラムから切り離すことで、検知方法の変更やバージョンアップなどに対応しやすくなる効果がある。

【００８５】次に本発明の第８の実施形態について図１０を参照して説明する。

【００８６】この実施形態では、改ざん検知機能３を持つ改竄検知プログラム９を補助記憶装置４に格納し、ＲＯＭ１内のブートプログラム２に、改ざん検知プログラム９を予め一方向性関数で圧縮した改ざん検知プログラム圧縮値３９１を記録しておく。そしてブート時には、現在の被検査プログラムの内容を一方向性関数で圧縮して値と、保存された改ざん検知プログラム圧縮値３９１

とを比較して、改ざん検知プログラム９自身の改ざんの有無を検査する（改ざん検知プログラム検査手段３９）。改ざん検知プログラム９を読み込む方法は、例えば改ざん検知プログラム９を記憶装置９１の所定の物理的位置から始まる物理的連続領域に格納しておき、その所定の位置からＢＩＯＳを使ってＲＡＭ１０２に読み出せばよい。改ざん検知プログラム実行手段３８は、ジャンプ命令でＣＰＵ１００のＩＰアドレスを、ＲＡＭ１０２に読み込まれた改ざん検知プログラム９の先頭アドレスに設定すればよい。

【００８７】この実施形態では、本発明を実施するためにＲＯＭ１に格納されたブートプログラム２が新たに備える機能が、一方向性関数を用いた簡易な改ざん検知プログラム検査手段３９と、改ざん検知プログラム９をロードして実行する改ざん検知プログラム実行手段３８だけであるので、ブートプログラム２のプログラムサイズを小さくでき、占有するＲＯＭ１のメモリ空間を小さくできる効果がある。

【００８８】次に本発明の第９の実施形態について図１１を参照して説明する。

【００８９】この実施形態では、改ざん検知機能３を備えた改ざん検知プログラム９を補助記憶装置４から読み込み実行する改ざん検知プログラム実行手段４６１と、補助記憶装置４内のブートセクタ以外の物理的位置に別途保存されたＯＳローダ４３を読み込み実行するＯＳローダ実行手段４６２とを備えた代替ＯＳローダ４６を、補助記憶装置４のブートセクタに持たせる。

【００９０】改ざん検知プログラム実行手段４６１は、例えば、改ざん検知プログラム９を補助記憶装置４の所定の物理的位置から始まる物理的連続領域に格納しておき、その所定の位置からＢＩＯＳを使ってＲＡＭ１０２に読み込み、ジャンプ命令でＣＰＵ１００のＩＰアドレスを、ＲＡＭ１０２に読み込まれた改ざん検知プログラム９の先頭アドレスに設定すればよい。

【００９１】ＯＳローダ実行手段４６２も同様に構成できる。つまり、別途保存したＯＳローダも補助記憶装置４の所定の物理的位置から始まる物理的連続領域に格納しておき、改ざん検知プログラム実行手段４６１と同様に読み込み、実行すればよい。

【００９２】この実施形態では、補助記憶装置４に複数のＯＳをインストールして使い分け使用環境において、起動するＯＳ、つまりは起動するブートセクタが指定された後に改ざん検知機能３が動作することを特徴とする。これにより起動するＯＳに必要なファイルだけ、改ざんの検査をさせることができ、検査処理時間の短縮が図れる。

【発明の効果】第１の効果は、ＯＳローダやＯＳが起動する前に、ＯＳローダやＯＳファイル、あるいはその他任意のファイルの改ざんの有無を、確実に、しかも使用者は意識せずに、検証することができることである。そ

の理由は、上記の検証を行なうのはブートプログラムであり、当該コンピュータシステム起動時には確実に実行されるものであり、しかもＲＯＭに格納されているのでこれ自身が改ざんされる心配はないためである。

【００９３】第２の効果は、ハードウエア構成が簡素に構成できることである。その理由は、通常システムが持つブートＲＯＭに改ざん検査を行なうソフトモジュールを追加することと、被検査ファイルを補助記憶装置４の物理的連続領域に配置し第１の保存情報４１を生成するユーティリティプログラムを使用するだけで、本発明を構成することができ、特別なハードウエアの追加を必須としないためである。

【００９４】

【図面の簡単な説明】

【図１】本発明の一実施形態の構成を示す図である。

【図２】本発明の動作を説明するためのフローチャートである。

【図３】第１の保存情報の一例を説明するための図である

【図４】第１の保存情報の他の例を説明するための図である。

【図５】本発明の他の実施の形態の構成を表わす図である。

【図６】第１の保存情報の他の例を説明する図である。

【図７】本発明の他の実施の形態の構成を表わす図である。

【図８】本発明の他の実施の形態の構成を表わす図である。

【図９】本発明の他の実施の形態の構成を表わす図である。

【図１０】本発明の他の実施の形態の構成を表わす図である。

【図１１】本発明の他の実施の形態の構成を表わす図である。

【図１２】従来の技術を説明するための図である。

【符号の説明】

１　ＲＯＭ
２　ブートプログラム
３　改ざん検知機能
３１　保存情報読み込み手段
３２　被検査ファイル読み込み手段
３３　改ざん検知手段
３４　第２の保存情報
３５　保存情報入力手段
３６　第２の保存情報読み込み手段
３７　通信手段
３８　改ざん検知プログラム実行手段
３９　改ざん検知プログラム検査手段
３９１　改ざん検知プログラム圧縮値
４　補助記憶装置

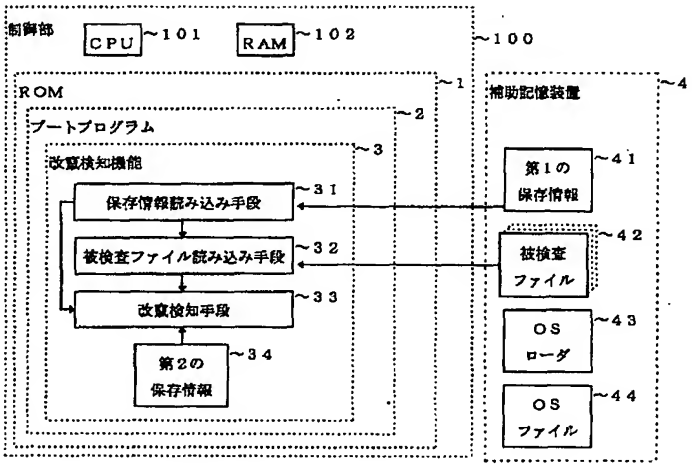
17

４１　第１の保存情報
４２　被検査ファイル
４３　ＯＳローダ
４４　ＯＳファイル
４５　ブートセクタ
４６　代替ＯＳローダ
４６１　改ざん検知プログラム実行手段
４６２　ＯＳローダ実行手段
４７　改ざん検知プログラム
４８　被検査ファイルリスト
６　入力装置

18

＊７　外部記憶装置
７１　改ざん防止手段
８　外部コンピュータ
８１　改ざん防止手段
８２　改ざん検知計算手段
９　改ざん検知プログラム
９１　ライトプロテクト機能付き記憶装置
１００　制御部
１０１　ＣＰＵ
１０２　ＲＡＭ
＊

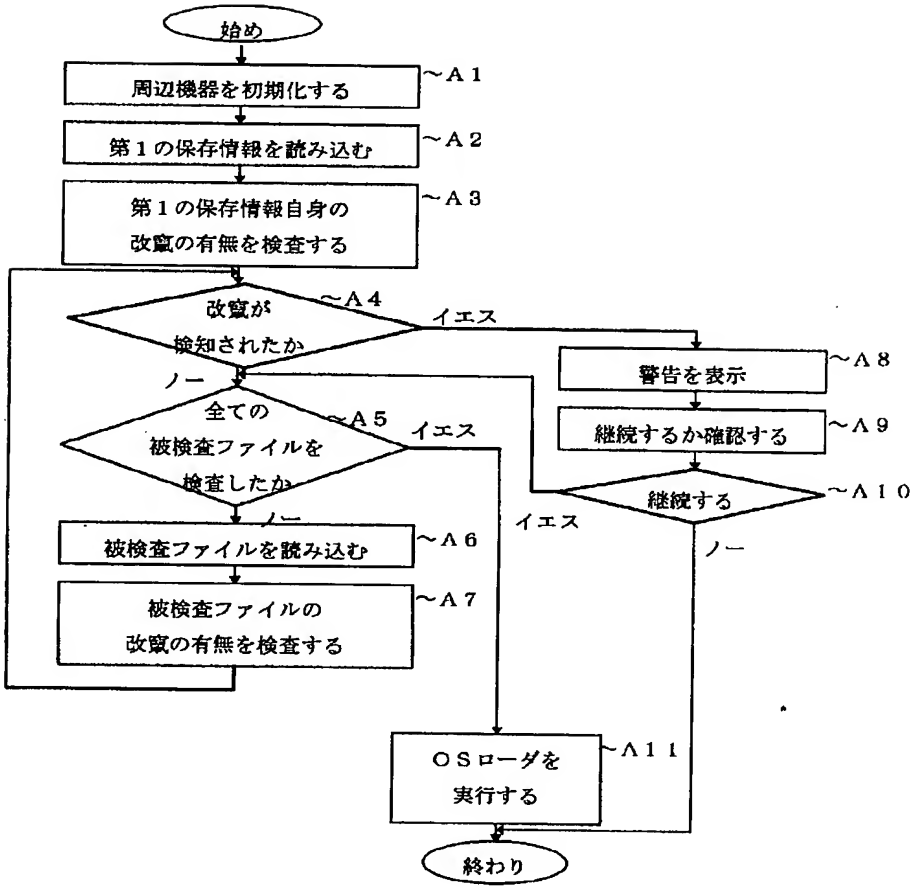
【図１】



【図３】

先頭マーカ	
全体サイズ	
被検査ファイル数（n）	
第１の被検査ファイル名	第１の被検査ファイルの署名データ
第２の被検査ファイル名	第２の被検査ファイルの署名データ
・・・・	・・・・
第ｎの被検査ファイル名	第ｎの被検査ファイルの署名データ
第１の保存情報自身の署名データ	

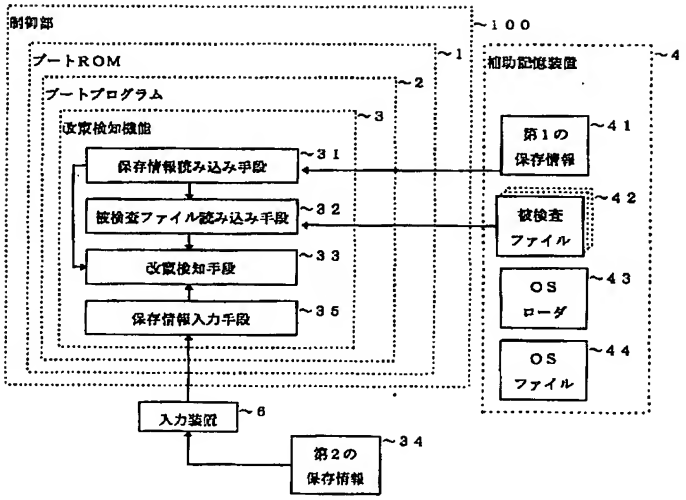
【図２】



【図４】

先頭マーカ	
全体サイズ	
被検査ファイル数（n）	
第１の被検査ファイル名	第１の被検査ファイルと第２の保存情報とを圧縮した圧縮値
第２の被検査ファイル名	第２の被検査ファイルと第２の保存情報とを圧縮した圧縮値
・・・・	・・・・
第ｎの被検査ファイル名	第ｎの被検査ファイルと第２の保存情報とを圧縮した圧縮値
第１の保存情報自身の署名データ	

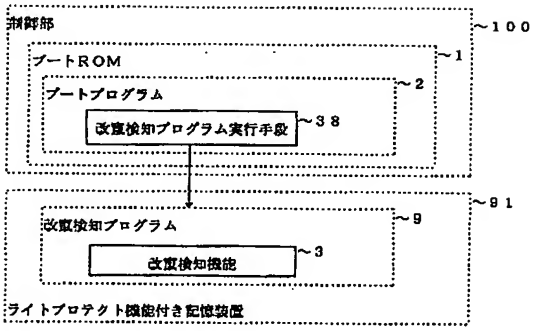
【図５】



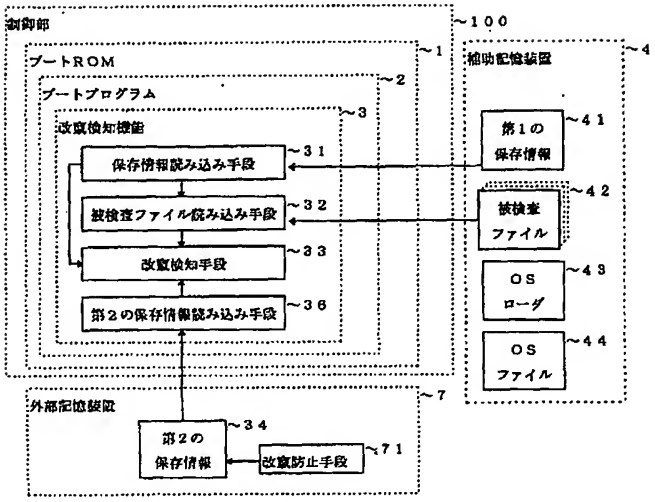
【図６】

先頭マーカ	
全体サイズ	
被検査ファイル数（n）	
第１の被検査ファイル名	第１の被検査ファイルの圧縮値
第２の被検査ファイル名	第２の被検査ファイルの圧縮値
・・・・	・・・・
第ｎの被検査ファイル名	第ｎの被検査ファイルの圧縮値
これより上の部分と第２の保存情報とを圧縮した圧縮値	

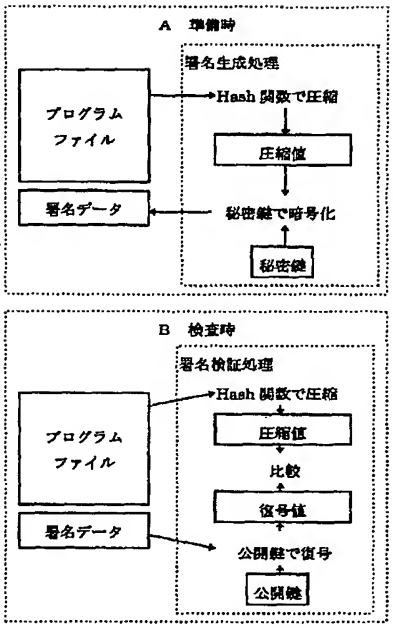
【図９】



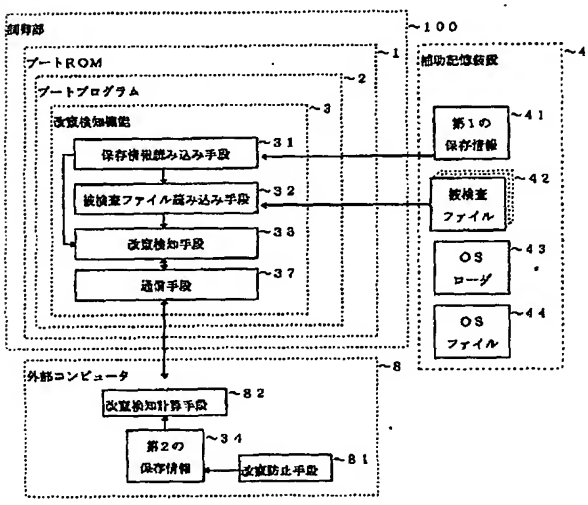
【図７】


制御部
～１００
～１
ブートROM
～２
ブートプログラム
～３
改竄検知機能
保存情報読み込み手段
～３１
被検査ファイル読み込み手段
～３２
改竄検知手段
～３３
第２の保存情報読み込み手段
～３６
補助記憶装置
～４
第１の保存情報
～４１
被検査ファイル
～４２
OS ローダ
～４３
OS ファイル
～４４
外部記憶装置
～７
第２の保存情報
～３４
改竄防止手段
～７１

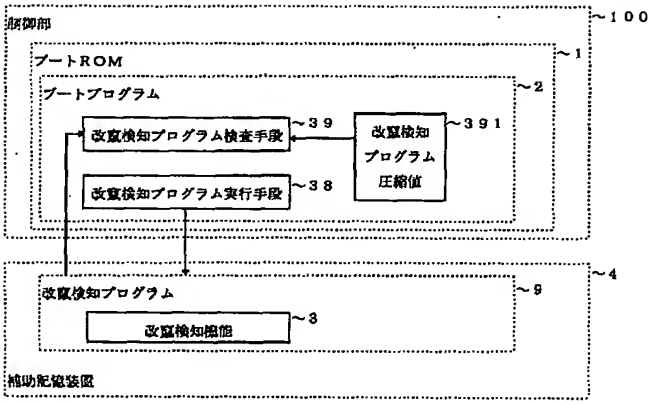
【図１２】


A　準備時
署名生成処理
プログラム ファイル
Hash 関数で圧縮
圧縮値
署名データ
秘密鍵で暗号化
秘密鍵
B　検査時
署名検証処理
プログラム ファイル
Hash 関数で圧縮
圧縮値
比較
復号値
署名データ
公開鍵で復号
公開鍵

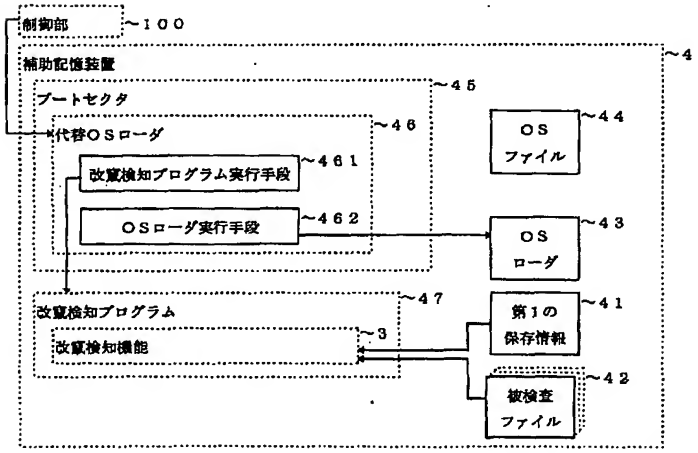
【図８】


制御部
～１００
～１
ブートROM
～２
ブートプログラム
～３
改竄検知機能
保存情報読み込み手段
～３１
被検査ファイル読み込み手段
～３２
改竄検知手段
～３３
通信手段
～３７
補助記憶装置
～４
第１の保存情報
～４１
被検査ファイル
～４２
OS ローダ
～４３
OS ファイル
～４４
外部コンピュータ
～８
改竄検知計算手段
～８２
第２の保存情報
～３４
改竄防止手段
～８１

【図１０】


制御部
～１００
ブートROM
～１
ブートプログラム
～２
改竄検知プログラム検査手段
～３９
改竄検知プログラム圧縮値
～３９１
改竄検知プログラム実行手段
～３８
～４
改竄検知プログラム
～９
改竄検知機能
～３
補助記憶装置

【図１１】


制御部
～１００
補助記憶装置
～４
ブートセクタ
～４５
代替ＯＳローダ
～４６
改竄検知プログラム実行手段
～４６１
ＯＳローダ実行手段
～４６２
ＯＳ
ファイル
～４４
ＯＳ
ローダ
～４３
改竄検知プログラム
～４７
改竄検知機能
～３
第１の
保存情報
～４１
被検査
ファイル
～４２